# Runtime Animation Generate Plug-in using Motion Diffusion Model

Daniel Cha[1], Haechan Je[2], Jaehoon Lee[3], Suho Park[4], and Yonghyeon Jo[5]

[1] College of Engineering, Advanced Materials Sciences and Engineering
[2] College of Engineering, Systems Management Engineering
[3] College of Engineering, Civil, Architectural and Environmental System Engineering
[4] College of Engineering, Mechanical Engineering
[5] College of Science, Physics

Sungkyunkwan University, Seoul, South Korea
Capstone Design Project 2023

**Abstract.** In this project, we present a novel approach to in-game 3D self-expression using a modified motion diffusion model in the Unreal Engine. Our goal is to create a plugin that enables users to easily and intuitively customize their character's animations in the game run-time by only writing some text. We describe our modified motion diffusion model and its implementation in Unreal Engine. Our approach allows users to generate diverse and natural-looking animations with minimal effort and save them for future uses, enhancing their immersive gaming experience. We present simple demo game that demonstrate the effectiveness and efficiency of our approach, showing that users can create and save personalized animations using our plugin. This work provides a significant contribution to the field of in-game character animation and self-expression, and we believe it has the potential to revolutionize the way users interact with their characters in the game run-time.

**Keywords:** Motion diffusion model · Unreal engine · Game run-time · Personalized animations.

## 1 Introduction

### 1.1 Motivation

Video games have evolved significantly in recent years, offering players more immersive and personalized experiences. One crucial aspect of these experiences is character animation, which plays a vital role in conveying personality, emotions, and actions of game characters. Traditional approaches to character animation in video games involve manual key-framing, motion capture, or procedural animation techniques. However, these methods have some limitations. Manual key-framing is a time-consuming and labor-intensive process that requires skilled animators, while motion capture can be expensive and require specialized equipment. Moreover, the number of animations that can be provided through these

methods is often limited, leading to repetitive and less diverse animations for characters in the game.

### 1.2 Proposal and Goal

Our proposal aims to address the limitations of traditional character animation techniques in games by introducing a plugin that leverages a modified motion diffusion model. Our goal is to provide users with a more efficient and effective way to create personalized animations for their in-game characters, ultimately enhancing the level of self-expression and immersion in gaming experiences.

By incorporating the motion diffusion model into the plugin, we offer users the ability to generate animations simply by typing text, eliminating the need for manual key-framing, motion capture, or specialized equipment. This intuitive and accessible approach empowers players of all skill levels to customize their character animations in real-time, fostering a sense of ownership and individuality within the game world.

Furthermore, our project seeks to overcome the limitations of repetitive and less diverse animations often observed in traditional methods. By utilizing the power of the motion diffusion model, we can generate a wide range of natural-looking animations that accurately convey personality, emotions, and actions of game characters. This diversity enhances the overall visual appeal and realism of in-game animations, leading to a more engaging and immersive gaming experience.

### 1.3 Technique and Approach overview

Our approach builds on the motion diffusion model, which is a powerful tool for generating motion using only text. The model is stored in a distinct server due to its heaviness. Text which is typed in from the user is sent to the server as an input for the model to create a motion data. The model sends the motion data back to the Unreal Engine as a response and we apply a dimension conversion to this data in order to convert it into a 'fbx' file format which can be used in the Unreal Engine. The plugin manager uses the converted data to create an animation and apply it to a character by adding a animation sequence. Lastly, the animation is saved in user's customized emote slot. By modifying the motion diffusion model and creating a plugin for the Unreal Engine, we aim to provide a more efficient and effective way to create personalized animations for characters in the game.

Also, this work provides a significant advancement in the field of in-game character animation and self-expression, overcoming many of the limitations of traditional animation techniques. We believe that our project has the potential to revolutionize the way users interact with their characters in real-time gaming, providing a more intuitive and personalized gaming experience.

## 2 Related Work

### 2.1 FLAME: Free-form Language-based Motion Synthesis & Editing

FLAME, Free-from Language-based Motion synthesis and Editing is the model that integration of diffusion-based generative models into the motion domain. FLAME can generate high-fidelity motions well aligned with the given text. Also, it can edit the parts of the motion, both frame-wise and joint-wise, without any fine-tuning. FLAME involves a new transformer-based architecture, devised to better handle motion data, which is found to be crucial to manage variable-length motions and well attend to free-form text. FLAME showed outstanding generation performances on three text-motion datasets: HumanML3D, BABEL, and KIT. Editing capability of FLAME can be extended to other tasks such as motion prediction or motion in-betweening, which have been previously covered by dedicated models.[1]

### 2.2 Deep Motion

Deep motion is an one example of a web application which uses machine learning and AI techniques for converting video to 3D animation into a game character. It uses AI motion capture technique, which captures and reconstructs full-body motion, including face and hand tracking. Deep Motion's AI-based approach to motion capture allows for highly accurate and natural-looking animations, making it an attractive option for developers looking to create high-quality character animations. With the increasing demand for immersive and realistic gaming experiences, the use of AI-based animation technology like Deep Motion is likely to become more prevalent in the game development industry.

### 2.3 Roblox

Roblox is an online game platform and game creation system developed by Roblox Corporation that allows users to program games and play games created by other users. Recently, Roblox integrated the Deep Motion's 3D animating technology and Vroid VRM Custom Character support which is a massive UI overhaul for easier and quicker animation creation. Users can utilize this in order to create a Roblox default game character by uploading .VRM file format character files which is used for AR/VR and diverse live streaming apps.

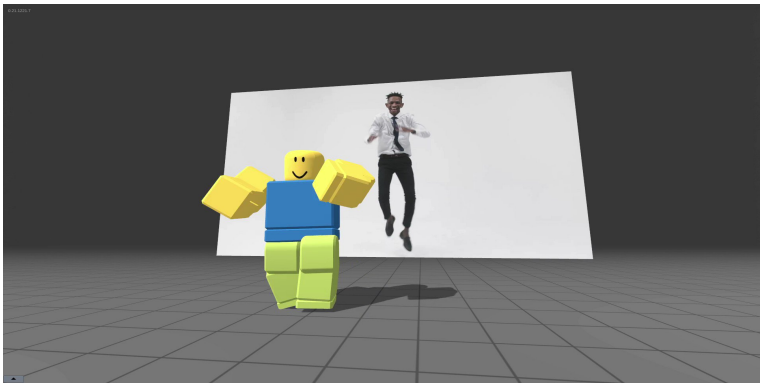

**Fig. 1.** Deep Motion applied in Roblox[2]

### 2.4 Kinetix

Kinetix is a company which developed and launched a no-code 3D creation tool powered by AI. It assures a democratized access to 3D animation for all creators by the user-friendly platform enabling User-Generated Content(UGC). The company aims to empower creator's self expression through emotes that can be used on any avatar in every virtual world and further interoperability with their developed game engine package.

The novelty of our project is generating animation with only text input in run-time game by integrating a motion diffusion model and a plugin for character animation. Our project stands out by offering users the ability to create diverse and natural-looking animations in real-time through a simple text input.

## 3 Proposed Service

### 3.1 Overall Structure

We provide motion generation API for Unreal Engine. In the Unreal Engine Plug-in, you can simply call the API to take over the fbx file needed for the character to animate. The following is what happens on the server when the API is called. The server first generates motion data and rotation data from the motion diffusion model. Unreal Engine uses rotation data, but the expressions used by motion diffusion models are quite different from those used by Unreal Engine. Therefore, we use a model called SMPLify to convert motion data obtained from the motion diffusion model into rotation data which is available at the unreal engine. Next, transform the smpl parameters obtained as a result of SMPLify into a format that fits the fbx file using fbx sdk. The Unreal Engine plug-in performs animation retargeting using the fbx file, and finally creates a motion that the user can use.

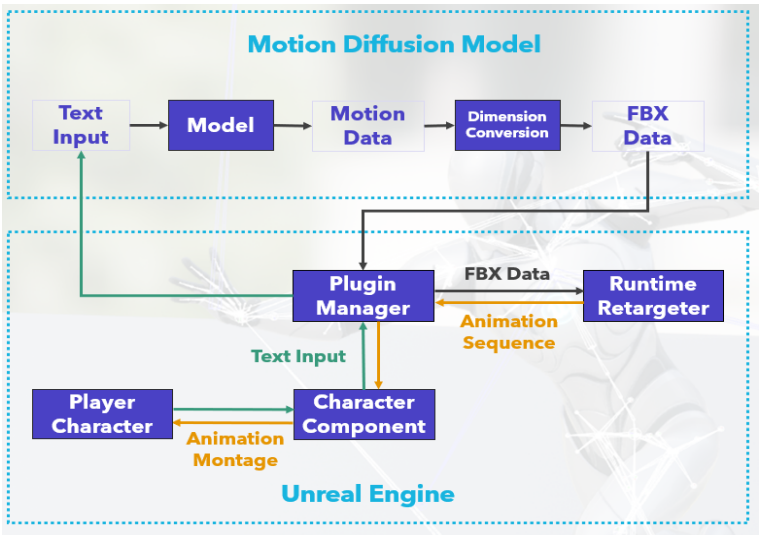


**Fig. 2.** Overall Pipeline of this project

### 3.2 MDM: Human Motion Diffusion Model

Motion Diffusion Model (MDM), is a diffusion-based generative model for the human motion. MDM is transformer-based, combining insights from motion generation models. A notable design-choice is the prediction of the sample, rather than the noise, in each diffusion step. This facilitates the use of established geometric losses on the locations and velocities of the motion, such as the foot contact loss. MDM is a generic approach, enabling different modes of conditioning, and different generation tasks. This model is trained with lightweight resources and yet produce excellent results on leading benchmarks for text-to-motion and action-to-motion. The MDM framework has a generic design enabling different forms of conditioning.[4]

### 3.3 Dataset

The HumanML3D dataset is a comprehensive and annotated collection of 3D human body models that serves as a valuable resource for advancing research in human pose estimation, 3D reconstruction, and related fields. It consists of a diverse range of human poses captured from multiple viewpoints and with varying levels of complexity. The dataset encompasses a large number of annotated 3D human body models, including information such as joint positions, skeletal connectivity, and mesh topology. A detailed description of the dataset acquisition process, the hardware setup, and the annotation methodology is employed to ensure accuracy and consistency. Due to the availability of the HumanML3D dataset, it is noticed to have a potential to facilitate the development and evalu-

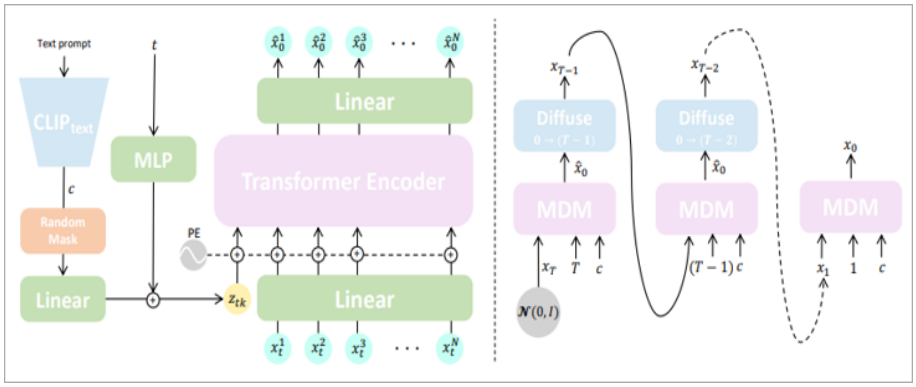


**Fig. 3.** Overall architecture of Motion Diffusion Model which is used for motion generation via text input[4]


**Fig. 4.** HumanML3D dataset: a dataset consisting of motion and a few text sentences about the motion[5]

ation of novel algorithms, enabling advancements in the understanding of human body pose estimation and related applications.

## 4 Implementation & Challenges

### 4.1 Model

#### Data Augmentation

We leveraged the HumanML3D dataset and the Motion Diffusion Model to generate emotion-related motions. However, we encountered a challenge related to the long tail problem, whereby there was an imbalance in the number of emotions represented in the dataset. To address this issue, we employed data augmentation techniques. By applying augmentation methods by replicating data with the most frequent emotion-related word replaced with a scarce word, we artificially increased the number of instances for the underrepresented emotions, effectively balancing the dataset. This approach allowed us to ensure a more representative distribution of emotions and enhance the robustness of our emotion-related motion generation system. This enabled us to train the Motion Diffusion Model on a more diverse range of emotional expressions, enhancing its ability to generate nuanced and realistic emotion-related motions.

#### Server

Another significant challenge we encountered was related to the computational cost of running the model locally on the Unreal Engine. The Motion Diffusion Model, being a computationally intensive model, demanded substantial resources that exceeded the capabilities of the client's hardware. To overcome this limitation, we implemented a client-server architecture. The client, operating within the game environment, sends a text input representing the text with desired emotion to a remote server. The server, which hosts the trained Motion Diffusion Model, processes the input and generates the created motion data. The generated motion data is then transmitted back to the client, allowing the game to incorporate emotion-related motions. By offloading the computational burden to the server, we were able to overcome the limitations imposed by the client's hardware and ensure the run-time generation of emotion-related motions in the game.

### 4.2 Unreal Engine

#### Connecting User Character to the Plugin

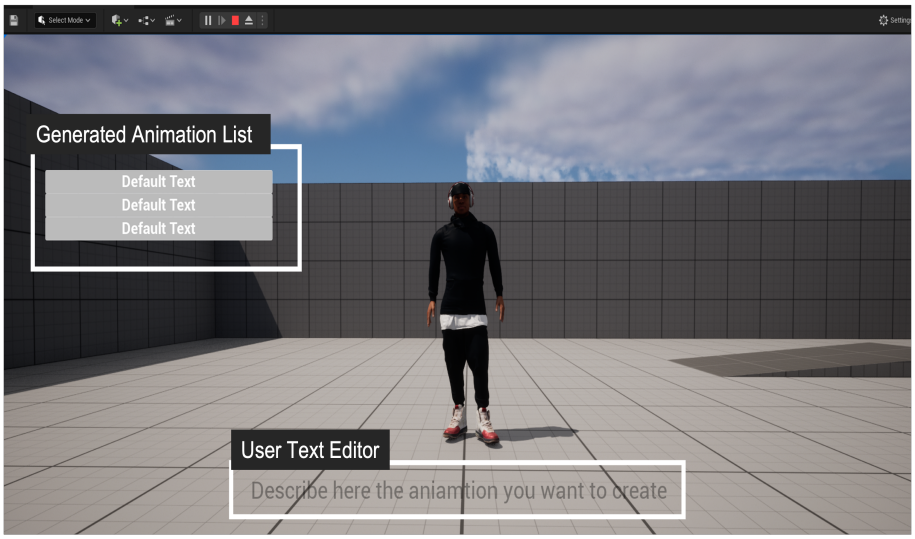


**Fig. 5.** UI for Runtime Animation Generate Plug-in on Unreal Engine

The first component of the plugin focuses on establishing a connection between the user character and the plugin. This involves integrating a text input mechanism that allows users to provide textual inputs within the Unreal Engine environment. The plugin receives this input and facilitates its transmission to the appropriate components for further processing. Additionally, the plugin enables users to specify the target character for animation application, thereby ensuring seamless integration between the user character and the plugin functionalities. Furthermore, the user character's skeleton information is transmitted to the plugin, providing the necessary data for animation retargeting.

### Retargeting Animations from a Model to the User Character

The second component of the plugin centers around the retargeting of animations from a model to the user character. This process involves extracting motion data from a model and acquiring animation information for each bone in the skeletal structure. To facilitate the transfer of animation information, the plugin incorporates a bone mapping mechanism, allowing the user character's skeleton to be matched with the SMPL (model) skeleton. Additionally, the plugin addresses differences in bone orientation across skeletons through advanced techniques such as Quaternion Rotation Multiplication. By effectively retargeting animations, the plugin empowers users to leverage pre-existing animation data for their user character, saving significant time and effort.

### Plugin Function(API) and Management

The third component of the plugin revolves around the definition and management of plugin functions through an API. To facilitate seamless communication and integration, the plugin incorporates server communication capabilities using asynchronous methods, such as Future and Promise. This ensures that the user interface remains responsive, even during server interactions. The API allows users to access a range of functions, including animation generation, storage, and playback. Notably, the PlayGeneratedAnim function enables users to provide text inputs, which are transmitted to the server for animation generation. Conversely, the GenerateAnimation function allows users to play animations generated based on the provided user text.

## 5 Evaluation

### 5.1 Fréchet Inception Distance (FID)

The Fréchet Inception Distance (FID) is a popular metric used to assess the quality and diversity of generated images produced by generative models, particularly Generative Adversarial Networks (GANs). The FID metric was introduced as an improvement over previous evaluation methods like Inception Score, which only measured image quality but not diversity. FID combines both aspects to provide a more comprehensive evaluation.

To compute the FID, a pre-trained model is utilized as a feature extractor. In motion generation work, motion feature extractor is used. Real and generated motions are passed through this model to extract feature representations. The features capture high-level information about the motion's content and style. The mean and covariance of these feature representations are then calculated for both the real and generated motion sets. The FID score is obtained by computing the Fréchet distance, a statistical measure of dissimilarity, between the mean and covariance of the real and generated features.

A lower FID score indicates that the generated motions closely resemble the real motions in terms of visual quality and diversity. It implies that the motion generative model has effectively learned the underlying distribution of the real data. Conversely, a higher FID score suggests that the generated motions significantly deviate from the real motions distribution, indicating poor quality or lack of diversity. Researchers and practitioners in the field of machine learning often utilize the FID metric to compare and evaluate different generative models, helping them make informed decisions about model performance and progress.

### 5.2 Evaluation Result

Our base model achieved an FID score of 0.544. However, due to computational constraints, we had to reduce the batch size for data augementation. As a result, we observed slightly higher FID score of 0.855. The obtained FID scores

| Model | FID |
|---|---|
| Original MDM Model | 0.544 |
| MDM Model + Augmentation (Ours) | 0.855 |
| SOTA model | 0.473 |

**Table 1.** FID results of original model, model with augmentation, SOTA

indicate that data augmentation has influenced the quality of the generated motions. While the original model achieved a lower FID score, the data augmentation(with reduced batch size), led to a higher FID score. This discrepancy can be attributed to the trade-off between computational resources and model performance. Despite the higher FID score, it is worth noting that data augmentation can still provide benefits such as increased diversity and robustness in the generated motions.

## 6 Limitation and Discussion

### 6.1 Computational Cost

Despite our efforts to address the computational demands of the Motion Diffusion Model (MDM) by deploying it on a remote server, we still encountered a limitation of extended execution times, since the MDM was renowned for its complexity and resource-intensive nature, resulting the limitation to achieving real-time performance. Consequently, the prolonged execution times adversely affected the responsiveness and user experience of our application, impeding the generation of emotion-related motions. Mitigating this limitation necessitates further exploration into optimization techniques and potentially considering alternative models or approaches that strike a balance between computational efficiency and motion fidelity.

### 6.2 Characteristic of dataset

Another limitation in our project was the confinement to generating motion specifically related to human-typed characters. This restriction stemmed from the limitations of the dataset we utilized, which lacked diverse character types beyond the human form which could have added diversity and richness to the motion. As a result, the generated motions were inherently biased towards human-like movements, limiting the range of motions that could be realistically produced. Furthermore, another significant limitation arose from the dataset's incomplete representation of hand bones. The absence of hand bone information within the dataset reduced the completeness and accuracy of the hand-related gestures and interactions. Future work should focus on expanding the dataset to include a broader range of character types and addressing the absence of hand bone information to enhance the completeness and diversity of motion generation capabilities.

## 7 Conclusion

In conclusion, our project aimed to revolutionize character animation in video games by developing a plugin that enables users to generate diverse and natural-looking animations using a modified motion diffusion model. By leveraging the power of text input, our approach provides an intuitive and personalized way for players to customize their character's animations in real-time, enhancing the immersive gaming experience.

We built upon the motion diffusion model, a powerful tool for generating motion using only text, and integrated it into the Unreal Engine through a plugin. This integration allowed users to create animations by simply typing in text, which was then processed by the model on a remote server. The generated motion data was converted into a usable animation format and applied to the user's character within the game. This process eliminated the need for manual key-framing or motion capture, offering a more efficient and accessible method for animation creation.

Throughout the development process, we encountered and addressed several challenges. We applied data augmentation techniques to balance the emotion-related motion dataset, improving the diversity and quality of generated animations. Additionally, we implemented a client-server architecture to overcome computational limitations and ensure real-time motion generation within the game environment.

Evaluation of our system using the Fréchet Inception Distance (FID) metric demonstrated its ability to generate high-quality and diverse animations. Although the computational time and character diversity presented some limitations, our project showcased the potential of using the motion diffusion model and plugin integration for in-game character animation and self-expression.

Looking ahead, future work should focus on optimizing the computational efficiency of the motion diffusion model, allowing for real-time performance. Expanding the dataset to include a broader range of character types and addressing the absence of hand bone information would enhance the versatility and realism of the generated animations. By continually improving and refining our plugin, we can unlock new possibilities for users to express themselves and create unique gaming experiences through personalized character animations.

## References

1. Kim, J., Kim, J., Choi, S.: Flame: Free-form language-based motion synthesis editing (2022).
2. DeepMotion youtube (2021.09), https://youtu.be/hPIO7DPLnUs. Last accessed 4 Oct 2017

3. Rospigliosi, Pericles ‘asher. ”Metaverse or Simulacra? Roblox, Minecraft, Meta and the turn to virtual reality for education, socialisation and work.” Interactive Learning Environments 30.1 (2022)
4. Tevet, G., Raab, S., Gordon, B., Shafir, Y., Cohen-Or, D., Bermano, A.H.: Human motion diffusion model (2022).
5. Guo, Chuan, et al. ”Generating diverse and natural 3d human motions from text.” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2022).
6. Loper, M.; Mahmood, N.; Romero, J.; Pons-Moll, G.; Black, M. J. SMPL: A skinned multi-person linear model. ACM Transactions on Graphics Vol. 34, No. 6, Article No. 248, (2015).